



Texas A&M University - Commerce
Department of Computer Science

Comparison of SVM and Random Forests for Heart Disease Risk Prediction

Lakshmi Chandana Narra

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

March 15, 2024

Declaration

I, Lakshmi Chandana Narra, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Lakshmi Chandana Narra
March 15, 2024

Abstract

In recent times, the global rise in cardiovascular diseases has become increasingly prevalent, influenced by evolving lifestyles and societal factors. Emphasizing the need for timely detection and ongoing monitoring, particularly in regions with limited medical resources. Utilizing a public health dataset on patient heart health, including information from medical procedures and ongoing patient monitoring, this research uniquely centers on the comparative analysis of SVM and Random Forests. Focused on these two algorithms, this research aligns with the evolving landscape of machine learning in healthcare, presenting a concentrated perspective on their potential contributions. The methodology involves training SVM and Random Forest models on the dataset, evaluating their performance using key accuracy metrics such as the confusion matrix, Accuracy, precision, and F1 score. The study anticipates achieving comparable accuracy between the models but aims to determine their relative strengths in precision, recall, and F1 scores. This research aims to provide insights into which algorithm may be better suited for addressing the challenges associated with cardiovascular health monitoring, taking into consideration all parameters assessed in the research conclusion.

Keywords: Machine Learning, Random Forest, Support Vector Machine, Cardiovascular disease , Healthcare

Acknowledgements

I extend my sincere gratitude to Professor Derek Harter, Ph.D., for his exceptional guidance and mentorship throughout this research project. His expertise and insightful feedback have been instrumental in shaping the direction of this study.

I would also like to acknowledge the Department of Computer Science at Texas A and M University-Commerce for providing a collaborative atmosphere and necessary resources for the successful completion of this research.

Special thanks to my friends and colleagues for their encouragement and valuable discussions during the various stages of this project.

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem statement	2
1.3	Aims and objectives	2
1.4	Solution approach	3
1.5	Summary of contributions and achievements	3
2	Literature Review	4
2.1	Review of State-of-the-art	4
2.2	Machine Learning : SVM and Random Forest	5
2.2.1	Machine Learning	5
2.2.2	Support Vector Machine	5
2.2.3	Random Forest	5
2.3	Critique of the review	5
2.4	Summary	6
3	Methodology	7
3.1	Dataset Description	7
3.1.1	Target Variable	8
3.1.2	Data Types	8
3.2	Software Used	9
3.3	Data preparation and cleaning	9
3.3.1	Dataset Loading and Data Type Definition	9
3.3.2	Handling missing values	9
3.3.3	Eliminate Duplicate Values	10
3.3.4	Data Preprocessing and Transformation	10
3.4	Example of a Figure in \LaTeX	11
3.5	Example of an algorithm in \LaTeX	12
3.6	Example of code snippet in \LaTeX	12
3.7	Example of in-text citation style	14
3.7.1	Example of the equations and illustrations placement and reference in the text	14
3.7.2	Example of the equations and illustrations style	14
3.8	Summary	15

4	Results	16
4.1	A section	16
4.2	Example of a Table in \LaTeX	17
4.3	Example of captions style	17
4.4	Summary	17
5	Discussion and Analysis	18
5.1	A section	18
5.2	Significance of the findings	18
5.3	Limitations	18
5.4	Summary	18
6	Conclusions and Future Work	19
6.1	Conclusions	19
6.2	Future work	19
7	Reflection	20
	Appendices	22
A	An Appendix Chapter (Optional)	22
B	An Appendix Chapter (Optional)	23

List of Figures

3.1	Top 10 patient data from dataset	9
3.2	Load data and data type definition	10
3.3	Example figure in \LaTeX	11

List of Tables

4.1	Example of a table in \LaTeX	17
-----	---	----

List of Abbreviations

SVM	Support Vector Machine
RF	Random Forest

Chapter 1

Introduction

Cardiovascular diseases represent a formidable global health challenge, with their prevalence escalating and ranking among the primary causes of morbidity and mortality. The pressing concern is the need for robust predictive models to address the increasing burden of heart diseases, enabling early detection and effective risk mitigation strategies.

1.1 Background

This research project delves into the development and evaluation of predictive models for heart disease using machine learning algorithms, specifically focusing on Support Vector Machines (SVM) and Random Forests(RF). The scope encompasses a comprehensive analysis of these algorithms, exploring their capabilities and limitations in accurately predicting the risk of cardiovascular events. The context of the project revolves around leveraging a diverse dataset derived from various medical procedures and continuous patient monitoring to enhance our understanding of heart disease prediction.

The background of this study lies in the evolving landscape of lifestyle, dietary habits, and healthcare dynamics that contribute to the increasing prevalence of cardiovascular diseases. The significance of early detection and continuous monitoring underscores the importance of advanced Toma and Wei (2023)predictive modeling techniques. Against this backdrop, the research aims to contribute to the field of cardiovascular health by providing insights into the efficacy of SVM and Random Forest algorithms.

In Kumari et al. (n.d.) a comparative study on classification methods namely Ripper, Decision Tree, Artificial neural networks and Support Vector Machine are analyzed on cardiovascular disease dataset.

In Yanwei et al. (2007)it is establishes that a number of factors have been shown to increase the risk of developing heart disease. Some of these family history, high levels of LDL bad cholesterol, Family history of cardiovascular disease, High levels of LDL (bad) cholesterol, Low level of HDL (good) cholesterol, Hypertension, High fat diet, Lack of regular exercise, Obesity.

In summary, the investigated problem centers on the escalating prevalence of cardiovascular diseases, and the project's scope involves the development and evaluation of predictive models using SVM and Random Forest. The background highlights the contextual relevance of advanced

predictive modeling in addressing the challenges posed by heart diseases in the contemporary healthcare landscape.

1.2 Problem statement

The research question guiding this study is: "How do Support Vector Machines (SVM) and Random Forest algorithms differ in terms of accuracy, efficiency, and interpretability when predicting the risk of heart disease?"

The prevalence of cardiovascular diseases is increasing globally, necessitating accurate and efficient predictive models for early detection and intervention. However, selecting the most suitable algorithm for this task poses a challenge. This research aims to compare and examine the differential performance of SVM and Random Forests in predicting the risk of heart disease. By leveraging real-world dataDavid Lapp (1988) on patient heart health, the study seeks to uncover the unique strengths and limitations of each algorithm. Through this investigation, the research aims to provide insights into selecting appropriate machine learning algorithms to enhance cardiovascular health monitoring and decision-making in clinical practice.

1.3 Aims and objectives

This research project's main goal is to evaluate and contrast the effectiveness of Random Forest and Support Vector Machines (SVM) algorithms in relation to risk assessments for heart disease. The main objective is to advance predictive modeling methods for accurate assessment and early identification of cardiac disease.

- Analyze and clean the Kaggle heart disease dataset, preparing it for building predictive models.
- Develop a Support Vector Machine (SVM) classifier and train it on the preprocessed heart disease data.
- Build a Random Forest (RF) classifier and train it on the same preprocessed data.
- Conduct parameter tuning to maximize the performance of both models, focusing on recall, precision, and/or F1 scores.
- Compare the recall, precision, and accuracy of the resulting SVM and RF models.
- Evaluate and interpret the performance of SVM and RF models in predicting heart disease risk.
- Identify the strengths and weaknesses of each model in the context of cardiovascular health monitoring.
- Draw conclusions and provide recommendations for selecting the most suitable machine learning algorithm for heart disease risk analysis.

1.4 Solution approach

Data Preparation: Prepare the Kaggle heart disease dataset for the construction of predictive models by analyzing and cleaning it.

Model Development: Using the preprocessed cardiac disease data, create an SVM classifier and train it. Using the same preprocessed data, create and train a Random Forest (RF) classifier.

Model Assessment: Optimize both models' performance by fine-tuning their parameters with an emphasis on recall, precision, and/or F1 scores.

Print the assessment measures for each of the two models:

- Metrics for the SVM Model: F1 Score, Accuracy, Precision, Recall.
- Metrics for the Random Forest Model: F1 Score, Accuracy, Precision, Recall.

Print the two models' confusion matrices.

Model Comparison: Based on the evaluation metrics and confusion matrices, compare the SVM and Random Forest models' performances. Analyze and assess how well the RF and SVM models predict the risk of heart disease. Determine each model's advantages and disadvantages in relation to cardiovascular health monitoring.

Concluding remarks and suggestions:

- Make inferences from the performance comparison.
- Make suggestions on which machine learning algorithm would be best for analyzing the risk of heart disease.

The measures used to answer the research topic of comparing and contrasting the efficiency of Random Forest and SVM algorithms for heart disease risk assessments are described in this solution methodology. Data preparation, model construction, assessment, comparison, and conclusion are all included, giving your research a thorough approach.

1.5 Summary of contributions and achievements

Chapter 2

Literature Review

This section explores the efficacy of machine learning (ML) in the prediction of cardiovascular disease (CVD). ML learns from data and experience through training, enabling it to be applied to various tasks based on specific algorithms. This flexibility enables ML algorithms to analyze complex datasets and predict CVD risk.

A review of existing literature is also done to investigate previously published studies in the area. This review helps to contextualize the current findings. The literature review for this study will look at earlier research on machine learning (ML) in disease prediction, including various algorithms and their effectiveness. This understanding is crucial for developing accurate and effective predictive models for CVD.

2.1 Review of State-of-the-art

Previous studies, such as that by Shedole and Deepika (2016), have focused on predicting chronic diseases by analyzing data from historical health records. They employed various techniques, including Naïve Bayes, Decision Trees, Support Vector Machines (SVM), and Artificial Neural Networks (ANN). Through a comparative study of these classifiers, the researchers evaluated their performance in terms of accuracy. Their findings indicate that SVM achieved the highest accuracy rate overall, while Naïve Bayes showed the best performance in predicting diabetes.

Shetty and Naik (2016) proposed the development of a predictive system for diagnosing heart disease using patient medical datasets. They considered 13 risk factors as input attributes for building the system. The data from the dataset was analyzed, and processes such as data cleaning and data integration were performed.

Pal and Parija (2021) In this study, the researchers implemented the random forest data mining algorithm to predict heart disease. Their experimental results showed a sensitivity of 90.6, specificity of 82.7, and an overall accuracy of 86.9 for heart disease prediction. The proposed system achieved a classification accuracy of 86.9 and a diagnosis rate of 93.3 using the random forest algorithm. The researchers suggest that the system could also be used for predicting other diseases by applying different machine learning algorithms such as Naïve Bayes, decision tree, K-NN, linear regression, and fuzzy logic to improve accuracy. They also propose the use of cloud computing technology to manage large volumes of patient data.

2.2 Machine Learning : SVM and Random Forest

2.2.1 Machine Learning

In the realm of artificial intelligence, machine learning is dedicated to developing statistical models and algorithms that enhance a computer's performance in specific tasks without the need for explicit programming. It revolves around utilizing statistical models and algorithms to execute tasks without direct instructions, relying heavily on pattern recognition and prediction.

2.2.2 Support Vector Machine

A supervised machine learning approach called Support Vector Machine (SVM) is commonly used for classification problems, while it can also be used for regression tasks. The way SVM works is that it finds the hyperplane in the input data set that best divides different classes. The margin, or the distance between the hyperplane and the nearest data point from each class also known as the support vectors is optimized when choosing this particular hyperplane. Because SVM uses a small amount of memory and performs well in high-dimensional spaces, it can be used to datasets with a large number of features.

2.2.3 Random Forest

In order to create the class that represents the mean prediction (for regression) or the mode of the classes (for classification), Random Forest is an ensemble learning technique that creates many decision trees during training. Each tree in a Random Forest is trained using a subset of the training set, and the ultimate prediction can be determined by polling (for classification) or by averaging the predictions of all the trees (for regression). High accuracy, scalability, and the capacity to handle big datasets with high dimensionality are attributes of Random Forest.

2.3 Critique of the review

The literature review offers insightful information about the application of machine learning algorithms more especially, SVM and Random Forest for the diagnosis and prognosis of cardiac disease. Shedole and Deepika (2016) revealed how SVM may be used to achieve high overall accuracy, while Naive Bayes showed encouraging findings in terms of diabetes prediction. This implies that SVM would work well for our research on the identification of heart disease, especially given its ability to handle high-dimensional data.

Shetty and Naik (2016) highlighted the significance of feature selection in machine learning models by proposing a prediction strategy for diagnosing heart disease based on 13 risk variables. This can be a useful case study for our work, highlighting the necessity of selecting relevant characteristics with consideration in order to increase the accuracy of our model.

Pal and Parija (2021) achieved a high sensitivity, specificity, and overall accuracy in their implementation of Random Forest for heart disease prediction. Their methodology demonstrates how ensemble approaches can enhance prediction performance, and we can take this into account when comparing Random Forest and SVM in our own study.

2.4 Summary

Overall, the reviewed studies provide a solid foundation and examples for our research on comparing SVM and Random Forest for heart disease detection. We can leverage their methodologies and findings to design our experiments, select appropriate features, and evaluate the performance of the algorithms effectively.

Chapter 3

Methodology

3.1 Dataset Description

The Kaggle heart disease dataset David Lapp (1988), which has 1025 samples with 14 attributes each, was used in this investigation. The dataset includes a number of clinical characteristics that are frequently used to determine whether a patient has cardiac disease. Each sample has the following characteristics and represents a patient:

1. Age (*age*): The age of the patient in years.
2. Sex (*sex*): The gender of the patient (1 = male, 0 = female).
3. Chest Pain Type (*cp*): The type of chest pain experienced by the patient, categorized into four types: typical angina (1), atypical angina (2), non-anginal pain (3), and asymptomatic (4).
4. Resting Blood Pressure (*trestbps*): The resting blood pressure of the patient in mm Hg.
5. Serum Cholesterol (*chol*): The serum cholesterol level of the patient in mg/dl.
6. Fasting Blood Sugar (*fbs*): The fasting blood sugar level of the patient, where 1 indicates a fasting blood sugar level greater than 120 mg/dl and 0 indicates a level less than or equal to 120 mg/dl.
7. Resting Electrocardiographic Results (*restecg*): The resting electrocardiographic results of the patient, categorized into three types: normal (0), having ST-T wave abnormality (1), and showing probable or definite left ventricular hypertrophy (2).
8. Maximum Heart Rate Achieved (*thalach*): The maximum heart rate achieved by the patient.
9. Exercise-Induced Angina (*exang*): Whether the patient experienced exercise-induced angina (1 = yes, 0 = no).
10. ST Depression Induced by Exercise Relative to Rest (*oldpeak*): The ST depression induced by exercise relative to rest.
11. Slope of the Peak Exercise ST Segment (*slope*): The slope of the peak exercise ST segment, categorized into three types: upsloping (1), flat (2), and downsloping (3).

12. Number of Major Vessels Colored by Fluoroscopy (*ca*): The number of major vessels colored by fluoroscopy, ranging from 0 to 3.
13. Thalassemia (*thal*): The thalassemia status of the patient, categorized into three types: normal (3), fixed defect (6), and reversible defect (7).
14. Target (*target*): The presence of heart disease in the patient, where 1 indicates the presence of heart disease and 0 indicates the absence of heart disease.

3.1.1 Target Variable

The target variable, *target*, indicates the presence of heart disease and is binary, where 1 represents disease present and 0 represents disease not present.

3.1.2 Data Types

The dataset consists of the following features with their respective data types:

- **age**: integer
- **sex**: integer
- **cp**: integer
- **trestbps**: integer
- **chol**: integer
- **fbs**: integer
- **restecg**: integer
- **thalach**: integer
- **exang**: integer
- **oldpeak**: float
- **slope**: integer
- **ca**: integer
- **thal**: integer
- **target**: integer

age	sex	cp	trestbps	chol	fb	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

Figure 3.1: Top 10 patient data from dataset

3.2 Software Used

The implementation and comparison of SVM and Random Forest models were performed using Python along with the following libraries:

- **Python:** Python programming language was used as the primary language for coding the models.
- **pandas:** The pandas library was utilized for data manipulation and analysis, including reading the dataset, creating dataframes, and structuring the data.
- **scikit-learn (sklearn):** The scikit-learn library was used for implementing the SVM and Random Forest algorithms, as well as for data preprocessing, model training, and evaluation.

These libraries provided the necessary tools and functions to effectively implement and compare the models, ensuring a robust and efficient analysis of the dataset.

3.3 Data preparation and cleaning

As an initial step for preparing and cleaning data in this study, a dataset comprising 1025 samples is read from a CSV file and transformed into a table using Python's pandas library. This procedure involves loading the dataset, associating the columns with their corresponding values, and constructing a table containing the samples. This process is crucial for structuring the data in an organized manner, which will simplify subsequent tasks such as data cleaning, managing missing values, and preparing the data for training and testing the SVM and Random Forest models.

3.3.1 Dataset Loading and Data Type Definition

To prepare the data for analysis, I first categorized each column based on its data type. Subsequently, I imported the dataset from a CSV file into a Pandas DataFrame. This step was crucial for organizing and analyzing the data efficiently.

3.3.2 Handling missing values

Next step in our data analysis is to check for missing values in the dataset and deciding on how to handle them. We used python to inspect the dataset for missing values to analyze and found that there were 0 missing values. Consequently, no values needed to be replaced. Initially, we

```

PS C:\Users\chand\desktop> python heart_code.py
age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
0 52 1 0 125 212 0 1 168 0 1.0 2 2 3 0
1 53 1 0 140 203 1 0 155 1 3.1 0 0 3 0
2 70 1 0 145 174 0 1 125 1 2.6 0 0 3 0
3 61 1 0 148 203 0 1 161 0 0.0 2 1 3 0
4 62 0 0 138 294 1 1 106 0 1.9 1 3 2 0
... ..
1020 59 1 1 140 221 0 1 164 1 0.0 2 0 2 1
1021 60 1 0 125 258 0 0 141 1 2.8 1 1 3 0
1022 47 1 0 110 275 0 0 118 1 1.0 1 1 2 0
1023 50 0 0 110 254 0 0 159 0 0.0 2 0 2 1
1024 54 1 0 120 188 0 1 113 0 1.4 1 1 3 0

[1025 rows x 14 columns]
age int32
sex int32
cp int32
trestbps int32
chol int32
fbs int32
restecg int32
thalach int32
exang int32
oldpeak float64
slope int32
ca int32
thal int32
target int32
dtype: object

```

Figure 3.2: Load data and data type definition

planned to fill any missing values with the mean, but since there were none, this step was not applicable in our data cleaning process.

3.3.3 Eliminate Duplicate Values

In the data cleaning phase, we detected and eliminated **723** duplicate entries from the dataset consisting of 1025 rows. This process was carried out using Python and the pandas library, which offers a `drop_duplicates()` method for identifying and removing duplicate rows while preserving the first occurrence of a duplicated row.

This step is crucial as duplicate entries can distort our analysis and lead to inaccurate findings. By eliminating duplicates, we ensure the consistency and accuracy of our dataset, enhancing its reliability for subsequent data analysis and modeling endeavors.

3.3.4 Data Preprocessing and Transformation

In this section, we describe the preprocessing and transformation steps applied to the raw dataset to prepare it for analysis and modeling.

Data Transformation

The dataset underwent several transformations to enhance its suitability for machine learning algorithms:

- **Normalization:** The columns 'age', 'trestbps', 'chol', 'thalach', and 'oldpeak' were normalized using the Min-Max scaling technique to a range of $[0, 1]$.

- **Standardization:** The same columns were also standardized using the StandardScaler to have a mean of 0 and a standard deviation of 1.
- **One-Hot Encoding:** Categorical variables including 'sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', and 'thal' were one-hot encoded to convert them into binary format.

The transformed dataset was saved into a single CSV file for further analysis and modeling.

3.4 Example of a Figure in \LaTeX

Figure 3.3 is an example of a figure in \LaTeX . For more details, check the link: wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.

Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the “Flow algorithm software”. Here is the link for that: flowgorithm.org.

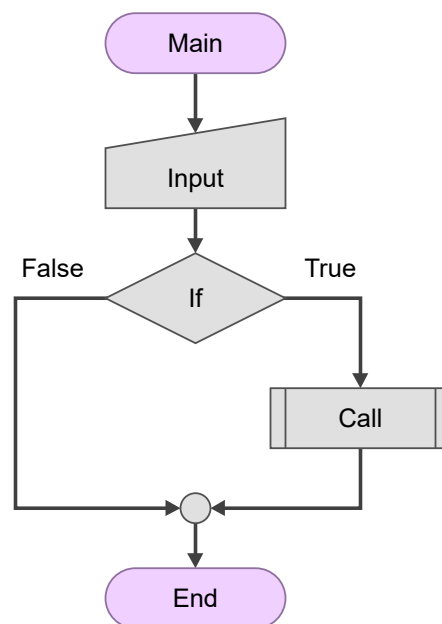


Figure 3.3: Example figure in \LaTeX .

3.5 Example of an algorithm in \LaTeX

Algorithm 1 is a good example of an algorithm in \LaTeX .

Algorithm 1 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \dots, x_N$

Output: *EvenSum* (Sum of even numbers in \mathbf{x})

```

1: function EVENSUMMATION( $\mathbf{x}$ )
2:   EvenSum  $\leftarrow$  0
3:    $N \leftarrow \text{length}(\mathbf{x})$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:     if  $x_i \bmod 2 == 0$  then                                ▷ check if a number is even?
6:       EvenSum  $\leftarrow$  EvenSum +  $x_i$ 
7:     end if
8:   end for
9:   return EvenSum
10: end function

```

3.6 Example of code snippet in \LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```

1 import numpy as np
2
3  $\mathbf{x}$  = [0, 1, 2, 3, 4, 5] # assign values to an array
4 evenSum = evenSummation( $\mathbf{x}$ ) # call a function
5
6 def evenSummation( $\mathbf{x}$ ):
7     evenSum = 0
8      $n = \text{len}(\mathbf{x})$ 
9     for  $i$  in  $\text{range}(n)$ :
10         if  $\text{np.mod}(\mathbf{x}[i], 2) == 0$ : # check if a number is even?
11             evenSum = evenSum +  $\mathbf{x}[i]$ 
12     return evenSum

```

Listing 3.1: Code snippet in \LaTeX and this is a Python code example

Here we used the “\clearpage” command and forced-out the second listing example onto the next page.

```

1 public class EvenSum{
2     public static int evenSummation(int[] x){
3         int evenSum = 0;
4         int n = x.length;
5         for(int i = 0; i < n; i++){
6             if(x[i]%2 == 0){ // check if a number is even?
7                 evenSum = evenSum + x[i];
8             }
9         }
10        return evenSum;
11    }
12    public static void main(String[] args){
13        int[] x = {0, 1, 2, 3, 4, 5}; // assign values to an array
14        int evenSum = evenSummation(x);
15        System.out.println(evenSum);
16    }
17 }

```

Listing 3.2: Code snippet in \LaTeX and this is a Java code example

```

1 int evenSummation(int x[]){
2     int evenSum = 0;
3     int n = sizeof(x);
4     for(int i = 0; i < n; i++){
5         if(x[i]%2 == 0){ // check if a number is even?
6             evenSum = evenSum + x[i];
7         }
8     }
9     return evenSum;
10 }
11
12 int main(){
13     int x[] = {0, 1, 2, 3, 4, 5}; // assign values to an array
14     int evenSum = evenSummation(x);
15     cout<<evenSum;
16     return 0;
17 }

```

Listing 3.3: Code snippet in \LaTeX and this is a C/C++ code example

3.7 Example of in-text citation style

3.7.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more than once.

3.7.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase “Eq” for an equation before using an equation number with (`\eqref{.}`). Use “Table” to refer to a table, “Figure” to refer to a figure, “Algorithm” to

refer to an algorithm and “Listing” to refer to listings (code snippets). Note that, we do not use the articles “a,” “an,” and “the” before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence “A report structure is shown in **the** Table ??” should be written as “A report structure is shown **in** Table ??.”

3.8 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter “**Testing and Validation**” should precede the “Results” chapter. See Section ?? for report organization of such project.

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in \LaTeX

Table 4.1 is an example of a table created using the package \LaTeX “booktabs.” do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of lines (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in \LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork)** goes **below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.3.
- The **caption of a Table** goes **above** the table. See the example in Table 4.1.
- The **caption of an Algorithm** goes **above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing** goes **below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with “Results” Chapter as “ Results and Discussion” as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader’s understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summarizes the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most significant contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in \LaTeX and use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planned aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is “**References**,” which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between “References” and “Bibliography.” The list of “References” strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of “References.”

References

David Lapp (1988), 'Public health dataset for heart disease prediction'.

URL: <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>

Kumari, M., Godara, S. and Kinoshita, A. (n.d.), 'Comparative study of data mining classification methods in cardiovascular disease prediction', *IJCST* **2**, 304–308.

Pal, M. and Parija, S. (2021), 'Prediction of heart diseases using random forest', *Journal of Physics: Conference Series* **1817**, 012009.

Shedole, S. S. and Deepika, K. (2016), Predictive analytics to prevent and control chronic disease, *in* 'Proceedings of the Conference Name'.

URL: <https://www.researchgate.net/publication/316530782>

Shetty, A. A. and Naik, C. (2016), 'Different data mining approaches for predicting heart disease', *International Journal of Innovative in Science Engineering and Technology* **5**, 277–281.

Toma, M. and Wei, O. C. (2023), 'Predictive modeling in medicine', *Encyclopedia* **3**(2), 590–601.

Yanwei, X., Wang, J., Zhao, Z. and Gao, Y. (2007), Combination data mining models with new medical data to predict outcome of coronary heart disease, *in* 'Proceedings International Conference on Convergence Information Technology', pp. 868–872.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...